QoCIM: A Meta-model for Quality of Context

Pierrick Marie, Thierry Desprats, Sophie Chabridon, Michelle Sibilla

QoCIM: A Meta-model for Quality of Context

Pierrick Marie[1], Thierry Desprats[1], Sophie Chabridon[2], and Michelle Sibilla[1]

[1] IRIT UMR 5505 Université Paul Sabatier, 31062 Toulouse, France
`<prenom>.<nom>@irit.fr`
[2] Institut Mines-Télécom/Télécom SudParis, CNRS UMR 5157, 91011 Évry, France
`Sophie.Chabridon@telecom-sudparis.eu`

Abstract. In the last decade, several works proposed their own list of quality of context (QoC) criteria. This article relates a comparative study of these successive propositions. The result is that no consensus has been reached about the semantic and the comprehensiveness of QoC criteria. Facing this situation, the QoCIM meta-model offers a generic, computable and expressive solution to handle and to exploit any QoC criterion within distributed context managers and context-aware applications. For validation purposes, QoCIM is successfully applied to the modelling of a set of simple and composite QoC criteria.

Keywords: Quality of context, Quality criterion, Context management, Meta-modelling, Information model.

1 Introduction

The expansion of the Internet of Things (the extension of the Internet to objects of the real world), cloud computing, big data and mobile technologies foster the development of new ubiquitous, context- and situation-aware applications. These situations are computed from ambient data, profiles of users and information collected from heterogeneous and spatially distributed sources. Context-aware applications become more and more usual. These applications require a fine and efficient management of the quality of the context information (QoC) they rely on. QoC is related to any information that describes the quality of context data as stated by the seminal definition of the QoC proposed by [4]. QoC specializes the general notion of Quality of Information (QoI) for context information.

A relevant behaviour of these applications strongly depends on the QoC provided. However, according to the business objectives of these applications, some QoC criteria may appear more important than others. Sometimes the freshness criterion is sufficient, sometimes it is the precision criterion and other times both are necessary. A solution to handle this need is to use context managers. They support context information throughout their life cycle. The life cycle of a piece of context information begins at its creation by a sensor and ends at its consumption by a context-aware application. Between these two events, context information are aggregated, filtered, deduced or transformed many times [3]. But these information are intrinsically incomplete and inaccurate [8]. A bad quality of context information could lead to wrong decisions and irrelevant reactions. That is why context managers must take into account QoC at each step of the context information life cycle. This challenge logically remains in the case of the next generation of multi-scale distributed context managers.

Several solutions have already been proposed. In 2007, the AWARENESS project [15] proposed a middleware to manage context information and offered a way to manipulate the QoC. In 2009, the COSMOS project [1] proposed mechanisms for the efficient management of QoC. Finally, one of the objectives of the INCOME project [2], started in 2012, is to provide context management solutions able to handle QoC as well as to preserve privacy.

Our objective is to provide future context managers with a *generic*, *computable* and *expressive* way to manipulate and exploit QoC simply and efficiently. *Generic*, because our solution has to model complex and heterogeneous QoC criteria. *Computable*, because the estimation of a quality level of a context information is based on treatments and operations on QoC criteria. Lastly, *expressive*, because context-aware applications must be able to express their QoC requirements to different context managers.

This paper is organized as follows. Section 2 compares the lists of QoC criteria that have been proposed over the last decade. After finding no standard list of criteria to measure QoC, we analyse in Section 3 different models able to bring a *generic*, *computable* and *expressive* solution to manipulate and exploit QoC. This led us to propose the QoCIM meta-model that we introduce in Section 4. Finally, Section 5 shows an instantiation of our meta-model, at design time, for a geolocation application and Section 6 concludes this paper.

2 Comparative Study on Proposed QoC Criteria Lists

We study in this section the existing works about QoC measurement. QoC measurement is based on a list of QoC criteria. Many authors have already established their own list of QoC criteria to measure QoC. We first enumerate the main proposals published over the last decade, and finally we compare the proposed criteria in terms of their semantic. The study highlights the existing variations in terms of denomination and meaning of QoC criteria. Different authors define a same meaning but associate it with a different denomination. On the contrary, a same denomination defined by different authors may correspond to different meanings.

2.1 Overview of QoC Criteria Lists

Buchholz 2003 [4] proposed in 2003 the first list of QoC criteria. This list is composed of five criteria : *precision*, *probability of correctness*, *trust-worthiness*, *resolution* and *up-to-dateness*. All of them are defined by a textual description. No computation method is formulated for their estimation, but Buchholz provides examples to illustrate each of them.

Kim 2006 [10] proposed in 2006 a new list of QoC criteria to measure the QoC. This list was built by confronting QoC criteria listed in [4] to generic criteria to measure quality. [10] provided five criteria associated to a definition from the point of view of the end users of the context information. The end user is the last entity which consumes context information. The proposed criteria are *accuracy*, *completeness*, *representation consistency*, *access security* and *up-to-dateness*. Then, [10] defined a mathematical formula to estimate the value of their first two criteria : *accuracy* and *completeness*.

Sheikh 2007 [15] for the AWARENESS project, formulated in 2007 its own list of QoC criteria. These criteria are *precision*, *freshness*, *temporal resolution*, *spatial resolution*, and *probability of correctness*. Although these criteria are fully described verbatim, no method is provided to estimate their value. Like [4], [15] gave examples to illustrate the definitions of their criteria. The descriptions of the criteria adopt successively the points of view of the consumer and of the producer of the context information. Producers are entities that create and provide context information as sensors, while consumers are context-aware applications.

Filho 2010 [7] studied the lists of QoC criteria that had been previously proposed by [4], [10] and [15] and imagined a new list of QoC criteria. Filho redefined *up-to-dateness*, *sensitiveness*, *access security*, *completeness*, *precision* and *resolution* criteria. For each criterion, Filho offered an example to illustrate the notion which is measured. Filho also provided a mathematical formula or a sample of Java program that he used to estimate these criteria.

Neisse 2012 [12] suggested in 2012 to adapt the ISO standard used in metrology to define QoC criteria. He established that the concepts of *accuracy* and *precision* used as QoC criteria are just an approximative definition of the precision criterion used in metrology. In the same way, Neisse estimated that the concepts of *spatial resolution* and *temporal resolution* defined by [15] are just a redefinition of the ISO standard of precision applied to spatial and temporal information. Neisse suggested to measure the QoC with only two criteria: the *age* and the *precision* of the context information. The *age* is the elapsed time since the production of the information. The *precision* criterion applies the ISO standard of precision on other kind of information depending the needs of the application. So, this *precision* criterion could be applied to the location of the source of the information, for example.

Manzoor 2012 [11] offered the most complete list of QoC criteria in 2012. They defined seven high level QoC criteria. All of them depend, for their computation, on other low level QoC criteria. For each of these high level QoC criteria, a mathematical formula is associated. The proposed criteria are *reliability*, *timeliness*, *completeness*, *significance*, *usability*, *access right*, *representation consistency*. The definition of some criteria adopts the point of view of the producer of the context information like the precision criterion provided by sensors. Whereas the definition of other criteria adopts the point of view of the consumer of the context information like context-aware applications which define the maximum allowed freshness of the received information.

2.2 Discussion

The study of the semantics of the QoC criteria listed above shows divergences. A same denomination of criterion appears in several lists with a different meaning. Conversely, a same meaning appears in many lists with different denominations. There are also meanings associated with denominations which appear only once into all the lists. Table 1 groups together the studied criteria by author and highlights the differences that exist between all of these criteria.

The different lists of QoC criteria are represented vertically. The name and the year of the first author of each list are mentioned on the first line of the table. The lists are sorted by publication date from left to right. Each criterion has a number, which is indicated in the first column of the table. The second column summarizes the meaning of each criterion. The cells of the table which contain a name, are criteria proposed by the authors registered on the top of the column of the cells. An empty cell indicates that the author did not propose the criterion referenced by the line of this cell. A cell with a check-mark represents a criterion implicitly used by the corresponding author but not clearly defined in its list of QoC criteria. Grey cells represent criteria defined by only one author. The lightgrey color indicates that there is one common meaning used by all authors. The criteria written in italic are names used only once. The criteria written in bold are names used by at least two different authors with different meanings. Some name of criterion are followed by numbers. For example, on line 15, the reliability criterion defined by Manzoor [11] is followed by the numbers 1, 2, 3 and 4. These numbers reference the numbers in the first column and indicate that this criterion is composed of other criteria. For example, Manzoor's reliability criterion [11] is computed using the first four criteria listed in this table.

Lastly, QoC criteria are sorted in the table by following a specific order. Criteria extracted directly from raw sensor data and which do not need computation or statistical analysis are placed on the top of the table. Whereas criteria at the bottom of the table require historical analysis or data from many sensors to be estimated. The more a criterion requires computations and lots of data, the lower it is placed in the table. Manzoor [11] classifies criteria into two categories, objective and subjective criteria; an objective criterion does not depend on the final application whereas a subjective criterion depends on the purpose of the final application. Table 1 rather orders criteria as a function of the effort that is required to estimate them.

Table 1 highlights that there is no consensus about which QoC criteria have to be used to measure the QoC of context information. This supports the idea of [3] indicating that a consensus about the definition of a common list of QoC criteria is still an open problem. Also the table provides a way to compare different lists of QoC criteria. This makes it possible to compare new specific lists between them. Indeed, with the development of context-aware applications, if a new high level criteria appear, Table 1 offers a method to classify lists of QoC criteria relatively to one another.

Table 1. Comparison of different lists of QoC criteria

		Buchholz 2003 [4]	Kim 2006 [10]	Sheikh 2007 [15]	Filho 2010 [7]	Manzoor 2012 [11]	Neisse 2012 [12]
1	Probability context is free of errors	**Correctness**	Accuracy		**Precision**	Accuracy	**Precision**
2	Max. distance to get context					*Sensor range*	
3	Location of the real world entity					*Entity location*	
4	Location of the sensor					*Sensor location*	
5	Time between production of contexts			*Temporal resolution*	✓	*Time period*	
6	Date of collection of context	✓	✓	✓	✓	*Measurement time*	*Timestamps*
7	Granularity location of context			*Spatial resolution*	**Resolution**		
8	Rate the confidence of the provider	*Trust worthiness*					
9	Critical value of context					*Significance*	
10	Granularity (detail level) of context	**Precision**		**Precision**	*Sensitiveness*	*Usability*	
11	Context consumer have access to context		✓			*Access right*	
12	Context transfers restricted, secured		Access security (11)		Access security		
13	Format respects consumer needs		Consistency			Consistency	
14	All aspects of entity are available	**Resolution**	Completeness		Completeness	Completeness	
15	Validity of context based on freshness	**Up to dateness** (6)	**Up to dateness** (6)	*Freshness* (6)	**Up to dateness** (5, 6)	*Timeliness* (5, 6)	
16	Believe in the correctness of context			**Correctness**		*Reliability* (1, 2, 3, 4)	

Meaning	Meaning used by all authors
Name	Name only defined by one author
Name	Criterion (name + meaning) only defined by one author
Name	Name defined by different authors for different meanings
Name	Name defined by different authors for the same meaning
Name (X)	The definition of this criterion depends on the X criterion
✓	Criterion not defined by author but another criterion depends on it

3 Study of Candidate Modelling Frameworks

Our objective is to provide a *generic*, *computable* and *expressive* modelling solution. Since no consensus can be reached about the list of QoC criteria that has to be used, the genericity of our solution cannot be based on a unique and exhaustive QoC criteria list. An alternative has to be found for enabling at least the cohabitation and ideally the integration of different lists of QoC criteria whose denominations, meanings and computation methods may differ. A solution to fulfil the genericity property is to provide either a common information model or a meta-model dedicated to quality of context. As far as we know, no such solution exists. Nevertheless, we decided to study several existing models, even if they address only partly our objective, in order to identify and possibly reuse some interesting concepts or modelling patterns they propose. Next sections discuss models of: the Open Geospatial Consortium, the IoT-A project, the COSMOS project, the DMTF and the Open Management Group.

3.1 Frameworks Study Overview

The Open Geospatial Consortium (OGC) model [14] represents observations made by sensors. It models the creation of context information when sensors observe and measure the real world. The model considers the process which is used to produce the result of an observation. It also associates a string with the result. This string contains the quality level of the result. However, the previous section shows that a quality level of a context information is based on a list of QoC criteria. A criterion is at least composed of a denomination, a meaning and a value. And a string is not enough to represent that. So, this model is not generic and expressive enough for our needs.

The IoT-A meta-model [9] provides methods and generic solutions to design applications based on the Internet of Things. The proposed meta-model defines the notion of attributes associated with meta-data. With this notion, meta-data could be considered as QoC criteria and fulfil our objective of a generic solution. However, the model does not include neither notion of context information or QoC criteria. It requires too much effort to extend and to exploit it correctly. The expressiveness and computability objectives are not fulfilled with this solution.

The COSMOS meta-model [5] expresses QoC management contracts. COSMOS is a context manager used for context-aware applications. It manages the QoC levels received by applications with a framework based on contracts. The meta-model proposed by COSMOS is used to establish very detailed quality contracts between a context-aware application and a context information provider. A contract defines the quality level that is required by an application. Such a level is defined for quality parameters, for a kind of context information. Through the definition of contracts, this meta-model fulfils our objective of providing an expressive solution. However, the computability property that we expect is not totally covered by this meta-model. The properties of a criterion are not clearly defined and prevent to build a generic and computable solution.

The DMTF CIM Metrics model [6] is one of the standards developed by the DMTF that is devoted to the management of metrics. It gives a way to express metrics that are used to qualify the state and the behaviour of managed system components (named a managed element) and how their respective values are obtained. In this sense, an analogy could be found with the qualification of context information. Context information are considered as managed elements. In this way, CIM Metrics are comparable to QoC criteria. The abstract class `ManagedElement` could represent context information which has to be qualified. The class `BaseMetricDefinition` that characterizes some metrics, could express a QoC criterion. The class `BaseMetricValue` that provides information on the valuation of an associated metric, could describe an estimation of a QoC criterion. Other subclass specialises metrics: the model proposes different computation patterns for metrics evaluation such as aggregation or discrete value selection. The genericity and computability aspects that we looking for could be fulfilled with the modelling technique used by CIM. It separates the definition of metric and the value of metric. The link between a QoC metric definition and its value is an association. It is a separation of concerns between, meanings and computation methods of QoC metric, and their values. This separation could be an early solution to obtain a generic and computable model of QoC criteria. However, this model does not cover the expressiveness aspect, there is no means for applications to express their requirements.

The Object Management Group (OMG) QoS meta-model [13] does not focus on the QoC domain but on the quality of service (QoS) domain. In this section, we highlight some common points between QoS and QoC management. Like for the CIM Metrics [6], the OMG approach really separates the definition of a criterion from its value. It organises the definitions of QoS criteria into categories to easily manipulate them. A lot of attributes are defined, among which the attribute `direction` compares different values of a same criterion from the point of view of the users of a service. For example, a service measures its response time; when this time increases, it means that the quality of this service decreases for the point of view of users. The attribute `unit` specifies the unit of a criterion, for instance using the units of the International System. The attribute `statisticalQualifier` specifies which statistical method is used to provide a value of a criterion. Another class is used to establish QoS contracts. QoS contracts are based on criteria represented by the class `QoSDimension`. The meta-model of the OMG offers at the same time the quality contracts just like COSMOS [5] and the separation of concerns between meanings and values used by the CIM [6] metrics. These two aspects cover our objective to provide a computable and expressive solution. However, the model does not supports the creation of new composite criteria depending on simple criteria.

3.2 Discussion

Among the studied models, none of them can easily provide without adaptation the three aspects of expressiveness, genericity and computability that we have identified as necessary. Table 2 summarizes the conclusions of our study of these

Table 2. Summary of the studied models

Model / Wished property	OGC	IoT-A	COSMOS	CIM	OMG
Expressiveness			✓		✓
Computability	✓			✓	✓
Genericity		✓		✓	

models. The closest models to our needs are the CIM Metrics model [6] and the QoS meta-model [13]. The next section introduces our proposed meta-model, QoCIM (QoC Information Model), which is inspired of these two models.

4 QoCIM : A New QoC Meta-model

QoCIM is our proposed meta-model able to design and to represent the QoC. It is not dependent on any QoC criterion. It offers a unified solution to model, at design time, heterogeneous QoC criteria. Then, models based on QoCIM could be used, at runtime, by both context managers and context-aware applications, for dynamic valuation of the QoC. This section describes the QoCIM meta-model.

4.1 Presentation of QoCIM

Figure 1 presents the QoCIM meta-model. QoCIM qualifies context information represented with the class `ContextInformation`. The quality of context

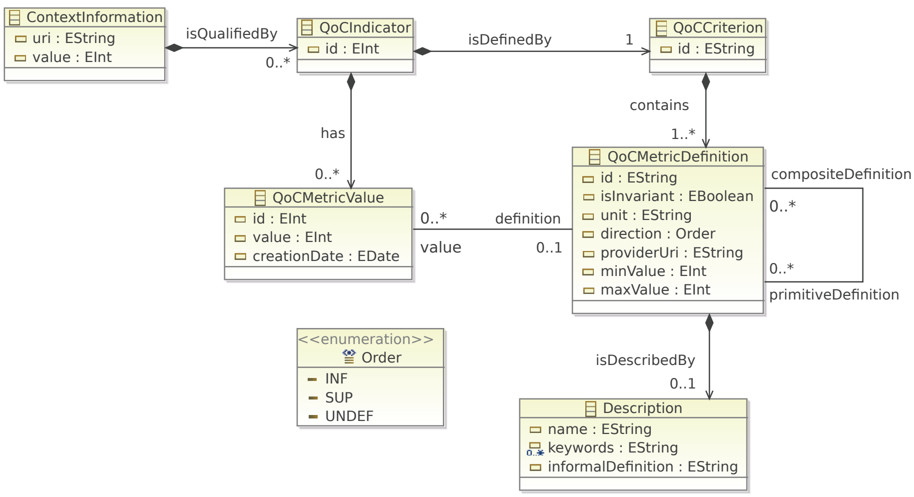

Fig. 1. QoCIM : QoC Information Model

information is designed with the `QoCIndicator`. An indicator is defined by one criterion, with the class `QoCCriterion`. Indicators and criteria are identified with the attribute `id`. At runtime, a valuation of the QoC is available with instances of the class `QoCMetricValue`. This class is identified with the attribute `id`. Its attribute, `value`, provides a valuation of the QoC. The date of creation of a value is contained into the attribute `creationDate`. The attributes of the class `QoCMetricDefinition` define the production of instances of `QoCMetricValues`:

- `isInvariant` indicates if the produced value is a constant, not editable, or dynamically computed.
- `unit` represents the unit of the produced value. It could be one of the units of the International System.
- `direction` compares different `QoCMetricValues` based on its attribute `value` from the point of view of the consumer of context information. The possible values of this attribute are INF, SUP and $UNDEF$:
  - INF means that a high `value` induces a better QoC level. For example, the freshness of a context information is usually computed with the following formula: $freshness = current\ date - date\ of\ the\ production\ of\ the\ context$ The result of this operation increases with the time whereas the quality of the information decreases.
  - SUP means that a high `value` induces a worth QoC level. For example, the precision of a context information computed with the following formula: $precision = 1 - \frac{distance\ between\ the\ sensor\ and\ the\ context}{maximum\ distance\ for\ the\ sensor\ to\ get\ context}$ More the sensor is close to the context, more the result of this operation and the quality of context increases.
  - $UNDEF$ is used when neither INF nor SUP can be expressed.
- `providerUri` identifies the resource that provides the `QoCMetricValue`. This attribute brings a way to filter the QoC based on the entity which computed it at runtime.
- `minValue` and `maxValue` respectively define the minimum and the maximum allowed value of the attribute `value` of the class `QoCMetricValue`.

The class `Description` brings semantics for the class `QoCMetricDefinition`. The attribute name contains the `name` of the description. The attribute `keywords` is a list of keywords. Finally, the attribute `informalDefinition` is a text that informally describes the `QoCMetricDefinition`. For the purpose of building composite criterion, the recursive association set on the class `QoCMetricDefinition` supports the ability to model and use a resulting criterion based on other criteria. Therefore, QoCIM authorizes `QoCMetricDefinition` depending on other classes `QoCMetricDefinition`.

4.2 Discussion

As the DMTF CIM metrics model [6] presented in Section 3, `QoCIM` separates the metric definition, `QoCMetricDefinition`, and the metric value, QoCMetricValue. QoCIM reuses a few attributes of the OMG QoS meta-model presented in Section 3 like isInvariant, direction and unit. QoCIM completes the attributes with providerUri and the class Description which are not specified in the OMG QoS meta-model. The DMTF CIM metrics model and the OMG QoS meta-model build higher level complex definitions of metric based on other definitions of metric. With the same objective, QoCIM also gives to designers of context-aware applications the ability to specify, new composite QoC criterion thanks to the recursive link set on the class QoCMetricDefinition. The next section presents an experimentation of QoCIM which is used at design time for defining three QoC criteria for a geolocation application

5 Experimentations

QoCIM is based on the EMF[1] technology. We used the EMF representation of QoCIM and the Obeo Designer[2] software tool to build a "QoCIM models editor". Obeo Designer is a tool that allows to quickly and easily develop editors of any instances of EMF meta-model. Thanks to the QoCIM models editor, we designed three QoC criteria for a geolocation application. The models of these criteria provide a definition of the QoC used for this application. These models are UML[3] class diagrams. Then, they could be exploited at runtime, to evaluate the QoC. Modelling these three QoC criteria followed two steps. The first step consisted in modelling two QoC primitive criteria. The first primitive criterion is the temporal resolution, the 5th criterion in Table 1. The second primitive criterion is the precision, the 1st criterion in Table 1. The second step consisted to represent a composite criterion based on the two criteria designed during the first step. This composite criterion is based on the temporal resolution and precision criterion.

5.1 Modelling the Temporal Resolution Criterion

The class diagram of Figure 2 shows the definition of the temporal resolution criterion. The value of the attribute id of the class TempResDefinition is "5.1", that is means TempResDefinition is the first definition of the fifth criterion of the Table 1. In this diagram, the default value of the attributes minValue is 0 and maxValue is 60 of the class TempResDefinition. The definition of this criterion is completed with the value of the attributes unit and direction which are respectively set to "minute" and INF. This criterion represents the elapsed time between the production of two context information. It means that more the value of this criterion increases more the quality of the context information decreases. The default values of the attributes of the class TemporalResolution present an informal description of this criterion.

[1] Eclipse Modeling Framework: www.eclipse.org/modeling/emf
[2] Obeo Designer: www.obeodesigner.com
[3] Unified Modeling Language: www.uml.org

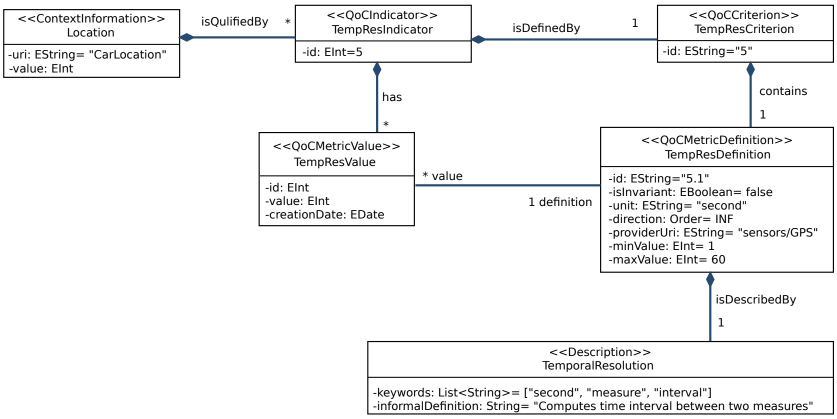

Fig. 2. QoCIM-based model of the QoC temporal resolution criterion

5.2 Modelling the Precision Criterion

The class diagram of Figure 3 shows the definition of the precision criterion. The value of the attribute `id` of the class `PerCentPrecDefinition` is "1.1", that is means `PerCentPrecDefinition` is the first definition of the first criterion of the Table 1. In this diagram, the default value of the attributes `minValue` is 0 and `maxValue` is 100 of the class `PerCentPrecDefinition`. The definition of this criterion is completed with the default value of the attributes `unit` and `direction` which are respectively set to "percent" and SUP. This criterion represents the estimation of the accuracy of the location. It means that more the value of this criterion increases more the quality of the context information increases. The default value of the attributes of the class `PerCentPrecision` presents an informal description of this criterion.

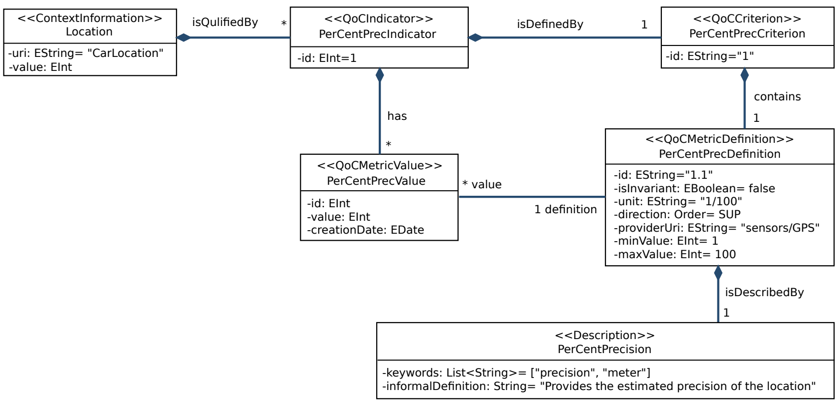

Fig. 3. QoCIM-based model of the QoC precision criterion

5.3 Modelling a Composite Criterion

Figure 4 presents the definition of a composite criterion. The composite criterion depends on the classes `PerCentPrecDefinition` and `TempResDefinition` designed previously. The `id` of this criterion is 17 because it could be classified into Table 1 as a new criterion, that is to say the seventeenth criterion. The value of the attribute `id` of the class `CompositeCriterion` is "$5.1 - 1.1$". This value refers to the value of the attribute `id` of the classes `TempResDefinition` which is "5.1" and `PerCentPrecDefinition` which is "1.1". The value of the attribute `id` of the class `CompositeDefinition` is "17.1", that is means `CompositeDefinition` is the first definition of the seventeenth criterion. This high level criterion may take three different `QoCMetricValue`s: `HighValue`, `MediumValue` and `LowValue`. These `QoCMetricValue`s are respectively associated to a default value: 1, 2 and 3. The production of these values are specified with OCL constraints. As an example, listing 1.1 shows the mandatory constraints to product an `HighValue`. As for the precision criterion, the value of the attributes `direction` of the class `CompositeDefinition` is SUP. It means that more the value of this criterion increases more the quality of the context information increase. The production of these values depends on the combined evaluation of the primitive criteria, precision and temporal resolution.

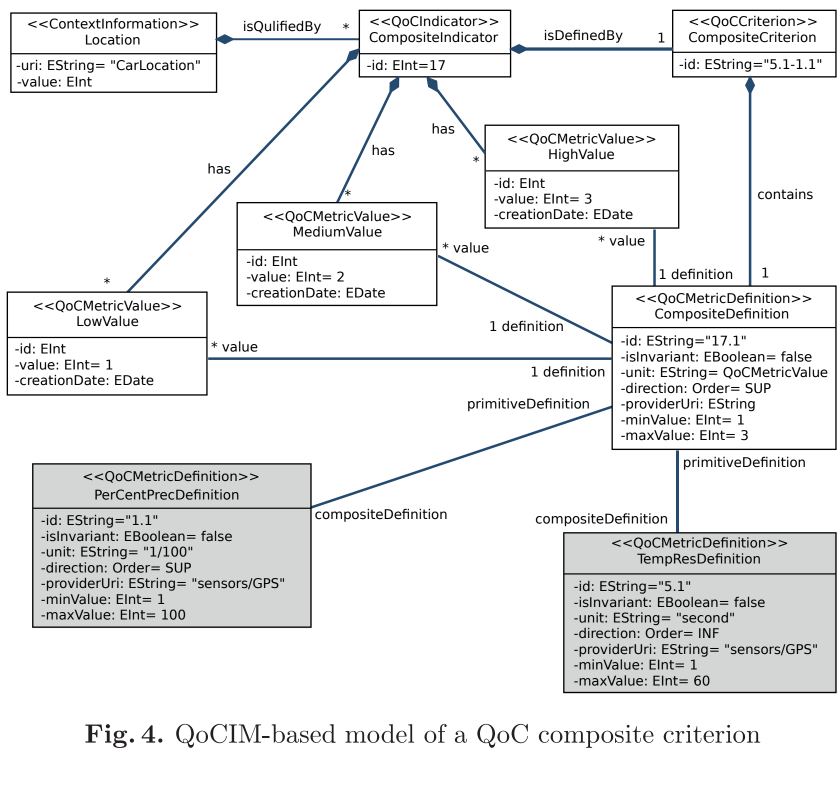

Fig. 4. QoCIM-based model of a QoC composite criterion

```
context CompositeDefinition::value(): HighValue
pre: self.PerCentPrecDefinition.QoCMetricValue.value >=
    90 % self.PerCentPrecDefinition.maxValue
pre: self.TempResDefinition.QoCMetricValue.value <=
    15 % self.TempResDefinition.maxValue
```

Listing 1.1. OCL constraints to define HighValue for the composite criteria

5.4 Discussion

The first step of the experimentation of QoCIM on the two primitive criteria of temporal resolution and precision, demonstrates that QoCIM is able to model low level criteria. We have also shown that QoCIM is able to model high level criteria derived from low level criteria. This can be applied to design more complex criteria like the granularity of a location context information, the 7$^{\text{th}}$ criterion of Table 1, or the trustworthiness of the provider, the 8$^{\text{th}}$ criterion. Figures 2 to 4 show that QoCIM can be used to model, at design time, in an unified way, the definition of any basic or composite QoC criterion. QoCIM is the same conceptual construct used to build those produced models. The processing at runtime any of these QoCIM-based models is then easier for context managers and context-aware applications when they have to deal with QoC criteria evaluation.

6 Conclusion and Perspectives

In the last decade, several works have addressed QoC modelling and management. This article presents the result of our analysis of existing modelling frameworks. Successive proposals of QoC criteria lists defined by different authors have been compared. The analysis explicitly demonstrates the existence of divergences and concludes on the difficulty to converge to a unique and exhaustive QoC criteria list. Facing this situation, we propose the meta-model, QoCIM. QoCIM is dedicated to exploit and to manipulate any QoC criterion within context managers and context-aware applications. It is built using the relevant concepts we have identified from other models dedicated or standardised for other domains. This article introduces the informational core of QoCIM. For validation purpose, QoCIM was successfully applied to the modelling of a set of simple and composite QoC criteria. Currently, we work on extending QoCIM and embedding new concepts able to express requirements on QoC criteria and QoC levels within context managers. The purpose of this extension is to offer for context managers a mechanism to specify and to control the QoC that context information producers supply and the QoC that context information consumers require. Thus, context managers will be able to evaluate QoC all along the life cycle of context information and apply filtering policies based on QoC.

Acknowledgments. This work is part of the French National Research Agency (ANR) project INCOME (ANR-11-INFR-009, 2012-2015).

References

1. Abid, Z., Chabridon, S., Conan, D.: A framework for quality of context management. In: Rothermel, K., Fritsch, D., Blochinger, W., Dürr, F. (eds.) QuaCon 2009. LNCS, vol. 5786, pp. 120–131. Springer, Heidelberg (2009)
2. Arcangeli, J.-P., et al.: INCOME – multi-scale context management for the internet of things. In: Paternò, F., de Ruyter, B., Markopoulos, P., Santoro, C., van Loenen, E., Luyten, K. (eds.) AmI 2012. LNCS, vol. 7683, pp. 338–347. Springer, Heidelberg (2012)
3. Bellavista, P., Corradi, A., Fanelli, M., Foschini, L.: A survey of context data distribution for mobile ubiquitous systems (2012)
4. Buchholz, T., Schiffers, M.: Quality of context: What it is and why we need it. In: Proceedings of the 10th Workshop of the OpenView University Association (2003)
5. Chabridon, S., Conan, D., Abid, Z., Taconet, C.: Building ubiquitous qoc-aware applications through model-driven software engineering. Science of Computer Programming (2012)
6. Distributed Management Task Force: Base Metric Profile (2009)
7. Filho, J.d.R.: A Family of Context-Based Access Control Models for Pervasive Environments. Ph.D. thesis, University of Grenoble Joseph Fourier (2010)
8. Henricksen, K., Indulska, J.: Modelling and using imperfect context information. In: Proc. 1st PerCom Workshop CoMoRea (2004)
9. Internet of Thing Architecture: Deliverable 1.3 (2012), `http://www.iot-a.eu/arm/d1.3` (last access May 2013)
10. Kim, Y., Lee, K.: A quality measurement method of context information in ubiquitous environments. In: Proceedings of the International Conference on Hybrid Information Technology (2006)
11. Manzoor, A., Truong, H.L., Dustdar, S.: Quality of context models and applications for context-aware systems in pervasive environments. Knowledge Engineering Review Special Issue on Web and Mobile Information Services (2012)
12. Neisse, R.: Trust and Privacy Management Support for Context-Aware Service Platforms. Ph.D. thesis, University of Twente, Enschede (2012)
13. Object Management Group: UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms Specification (2008), `http://www.omg.org/spec/QFTP/1.1/PDF` (last access May 2013)
14. Open Geospatial Consortium: Topic 20: Observations and Measurements (2007), `http://portal.opengeospatial.org/files/?artifact_id=41579` (last access May 2013)
15. Sheikh, K., Wegdam, M., Van Sinderen, M.: Middleware support for quality of context in pervasive context-aware systems. In: Pervasive Computing and Communications Workshops (2007)